

DCRM V4 搭建 Cydia Repo 教程

Dumb-Cydia-Repository-Manager

DCRM 介绍

DCRM – Darwin Cydia Repository Manager (V4) 是一个基于 Python Django 框架的 Cydia 源管理系统，对于新手来说旧版的基于 php 的 DCRM 安装会更加友好（网上也有很多相关安装搭建教程）。

但是旧版 WEIPDCRM 将于 2017 年 4 月起不再提供更新与技术支持，

目前新版的 DCRM4 在长期维护更新，并且采用了更美观的前台/后台以及更好的架构。

公众号 Cydiapp 希望人人都能用上最新的 DCRM-Cydia 源管理系统，搭建属于自己的 Cydia 源。

开源代码地址

<https://github.com/82Flex/DCRM>

<https://github.com/XLsn0w/Dumb-Cydia-Repository-Manager>

1、安装

centos 服务器，**宝塔面板安装** theos, git

2、Docker 和 Docker Compose 安装

<https://docs.docker.com/engine/install/centos/>

<https://docs.docker.com/compose/install/>

<https://www.runoob.com/docker/centos-docker-install.html>

3、DCRM 搭建

//只克隆下包含最近一次 commit 的一个分支

```
git clone --depth 1
```

```
git clone https://github.com/XLsn0w/Dumb-Cydia-Repository-Manager.git
&& cd Dumb-Cydia-Repository-Manager
```

也可以直接下载 zip，解压 cd 到文件夹路径下

编辑 docker-compose.yml 文件
可以根据需求修改端口映射

```

1 version: '3.2'
2
3 # do not change any thing if you do not know what it means
4 services:
5   web:
6     container_name: dcrm-web
7     hostname: dcrm-web
8     image: nginx:1.17.6
9     restart: always
10    ports:
11      - "8080:80" # test
12      - "80:80" # production
13      - "443:443" # https
14    volumes:
15      - app_static:/usr/share/nginx/html/static
16      - app_resources:/usr/share/nginx/html/resources
17      - ./docker/nginx/conf.d:/etc/nginx/conf.d # important: modify nginx c
18      - ./docker/nginx/certs:/etc/nginx/certs # your https certs: docker/n
19    depends_on:
20      - app
21      - worker-default
22      - worker-high
23    links:
24      - app:dcrm-app
25  pure-ftp:
26    container_name: dcrm-pure-ftp
27    hostname: dcrm-pure-ftp
28    image: williamlambert/pure-ftp:latest

```

修改挂载目录

前两个是匿名挂载，实际的目录可能是

/var/lib/docker/volumes/300c2264cd0acfe862507eedf156eb61c197720f69e7e9a053c87c2182b2e7d8/_data

由 docker 容器自己生成，一般不需要修改。

后面两个指定的是相对地址，也就是 docker-compose.yml 所在的目录下的地址

```

71   container_name: dcrm-worker-default
72   hostname: dcrm-worker-default
73   restart: always
74   build: .
75   deploy:
76     mode: replicated
77     replicas: 1
78   volumes:
79     - app_static:/usr/src/app/static
80     - app_resources:/usr/src/app/resources
81     - app_upload:/usr/src/app/upload
82     - app_temp:/usr/src/app/temp
83     - ./gnupg:/usr/src/app/.gnupg
84   links:
85     - db:dcrm-db
86     - redis:dcrm-redis
87     - memcached:dcrm-memcached
88   environment:

```

修改 replicas 为 1（这是应该是 bug，不修改会报错）

```

84 links:
85   - db:dcrm-db
86   - redis:dcrm-redis
87   - memcached:dcrm-memcached
88 environment:
89   PYTHONUNBUFFERED: 1
90   DCRM_WORKER: default
91 env_file:
92   - ./DCRM/.env
93   - ./docker/DCRM/.env
94 worker-high:
95   container_name: dcrm-worker-high
96   hostname: dcrm-worker-high
97   restart: always
98   build: .
99   deploy:
100     mode: replicated
101     replicas: 1
102   volumes:
103     - app_static:/usr/src/app/static
104     - app_resources:/usr/src/app/resources
105     - app_upload:/usr/src/app/upload

```

执行构建命令

```
docker-compose up --build --detach
```

构建完成之后就可以在浏览器上通过: <http://ip/admin> 进行访问

进入 dcrm_app 容器

```
docker exec -it affd3532b42a /bin/bash
```

修改 settings.py

```

apt-get update
apt-get install vim
vim settings.py

```

重启容器，稍微等一会，再次访问就进来了

创建超级用户

进入 dcrm_app 容器，创建账号，创建完成之后就可以登录了

```
cd /usr/src/app  
python manage.py createsuperuser
```

添加软件包

theos 项目目录中执行，

在 **packages** 目录下会生成 deb 文件，

将它上传上来就好了

```
make  
make package
```

将软件包设置为启用状态

全部完成之后就可以在手机上添加源：<http://ip>

在设置页面把之前添加的元数据设置为活跃状态：

（添加完源保存后页面导航条下面也有提示可以直接点击设置为活跃）

增删改查 debs

接着添加你的软件包：

上传 deb 格式的软件包：

启用软件包并保存：

最后构建活跃元数据：

详情任意填写注释信息即可：

注意：每次添加软件包后都需要重新构建一次，不然刷新失败，仍旧是上一次的提交数据